International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)
ISSN(P): 2249-6831; ISSN(E): 2249-7943
Vol. 5, Issue 2, Apr 2015, 97-102

TOR: PREVENT DNS LEAK THROUGH HONEYPOT

SUMAN GAUTAM[1], NITESH GUPTA[2] & SINI SHIBU[3]

[1]Department of CSE, NIIST (Affiliated to RGPV), Bhopal, Madhya Pradesh, India

[2,3]Asisstant Professor, Department of CSE, NIIST (Affiliated to RGPV,) Bhopal, Madhya Pradesh, India

ABSTRACT

Today, DNS/IP leakage in major problem of cyber crime world. In TOR, DNS are leaked easily in the time duration of whole communication between client and server and many users can't realise these types of hazard. So, attacker takes advantages of this system when it's reached the original IP of client and done much crime through other identity. In this paper we try to do new research in the world of cyber crime for rectified the problem. We are going to introducing the HONEYPOT in a TOR browser. Honey pot is a trap set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. Generally, a honey pot consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated and monitored, and which seems to contain information or a resource of value to attackers. This is similar to the police baiting a criminal and then conducting undercover surveillance.

KEYWORDS: TOR, HONEYPOT, DNS, PROXIES

INTRODUCTION

Today's, cyber crime or DNS/IP leakage is the major problems of communication system. Last few year ago launched zombie technique for preventing the system to cyber world. In zombie technique follow only the single path for established the communication between clients to server so, its major drawback is that attacker easily track the path and attacked to whole system. For the purpose of solving these problem launched TOR in the cyber crime world. In TOR communication is done between clients to server via random paths through multiple proxies. In TOR major problems are finding leaking the IP/DNS during the communication. In this paper we solve DNS leaking problem through Honey pot in TOR. We also describe TOR and Honey pot in below section. Literature is described in section 3 and then section 4 is problem finding section. We describe our proposed technique and algorithm in the section 5, and then finally describe result in section 6. Finally conclusion section is described in the last section 7.

TOR

TOR is stand for "THE ONION ROUTER". Generally TOR is used for hiding the identity. TOR communicate client to server via random path. The messages sent over an onion routing network were encrypted with their routing information and delivered to an intermediate server for forward delivery. Unlike the mix however, messages delivered using the onion routing network were encrypted multiple times, each 'layer' using a different encryption key and routing instructions[3]. The first node in a chain would only be able to encrypt the routing instructions to deliver the message to the next node. Each node in the sequence decrypting a layer until the complete message is decrypted and transmitted to the destination.

Honey Pot

Lance Spitzner, Founder of Honeypot technology. Honeypot is a resource that capture the aatacked by hacher/attacker. Honey can hope to detect, attack and even break; Honeypot is not employed to solve a problem, but it is mainly employed to collect valuable information about the attack[2]. Carefully set by the Honeypot system to attract hackers, and track to hacker the intruder can be observed record system.

Honeypot can be a computer simulation of a known hole or a service computer, also can simulate a variety of operating system and its corresponding features, or just a normal standard operating system, and only through special processing can be a complete record of the attacker's attack. Usually placed in a Honeypot want some of the intruder want sensitive information, of course, the information is false, or deliberately leave some security holes. The intruder and Honeypot the longer the interaction, they use techniques and methods on the more exposed, the more information gathered. This information can be used to understand the intruder to use the attack tools and methods to discovery of unknown holes and the corresponding attack, so you can better protect the systems and network security. In practice, Honeypot can be a process, a machine such as a bait to be deployed in their network to attract hackers to attack. Because they do not provide real value to the outside services, so any communication with their conduct is suspicious. As the attackers to spend time and effort on the Honeypot, while the real system will be protected security. In addition, Honeypot can also track the attacker can provide valuable clues.

LITERATURE REVIEW

R. Dingle din present TOR, he address the low latency communication services. He said Alice always uses two hops and then both onion routers can be certain that by colluding they will learn about Alice and Bob. Then he presents his approach where Alice always chooses at least three nodes unrelated to herself and her destination. [4]Should Alice choose a random path length (e.g. from a geometric distribution) to foil an attacker who employs timing to learn that he is the fifth hop and thus concludes that both Alice and the responder are running onion routers. Throughout in his paper, he has assumed that end-to-end traffic confirmation will immediately and automatically defeat a low-latency anonymity system. Even high-latency anonymity systems can be vulnerable to end-to-end traffic confirmation, if the traffic volumes are high enough, and if users' habits are sufficiently distinct [5, 6]. Can anything be done to make low-latency systems resist these attacks as well as high-latency systems? Tor already makes some effort to conceal the starts and ends of streams by wrapping long-range control commands in identical-looking relay cells. Link padding could frustrate passive observers who count packets; long-range padding could work against observers who own the first hop in a circuit. But more research remains to find an efficient and practical approach. Volunteers prefer not to run constant-bandwidth padding; but no convincing traffic shaping approach has been specified.

A. Chaabane M. Kaafar, **P. Manils, and** proposed a Tor exit node to detect, with a high probability, connections that are exploiting the exit node as a Tor tunnel. Once a Tor client builds a circuit, it sends specific Tor control messages which also called RELAY_BEGIN cells to instruct the last hop in the circuit for the exit node to establish a TCP connection to the destination host/port specified in the cell. Typically, the client *randomly* chooses three Tor nodes to build a three-hop circuit.

Hence, when the client sends the RELAY_BEGIN [7] cells, the chosen exit node receive the cells from a connection whose source's IP address belongs to the middle node. In the Tor tunnelling case, the client builds a one-hop

circuit, thus establishing a direct connection to an exit node, and it starts sending the RELAY_BEGIN cells. In other words, the problem of identifying Tor tunnels can be summarized in identifying connections carrying RELAY_BEGIN cells that do not originate from a Tor onion router.

Aaron Johnson and Paul Syverson proposed first set out a simple model that should facilitate reasoning about using trust in routing. He define trust simply to be the probability that an attempt by the adversary to control a node fails. He include a roving adversary that can attempt to compromise a certain number of nodes. [8]Route selection is modeled as a three-stage game in which the user first picks a distribution over paths, then the adversary chooses a set of nodes to attempt to compromise, and finally the user samples a path from his distribution. While we expect this model to bear further fruit, we use it in this paper to show a number of results of both theoretical and practical interest.

Murdoch and Watson [9] Conducted an empirical analysis that investigated the relationship between the path selection algorithm and path compromise with respect to the attacks cost for the adversary. They identified the fraction of malicious Tor routers and the fraction of adversary-controlled bandwidth as important factors for predicting the adversaries ability to compromise paths.

Browser based attacks on Tor was presented by **Abbott et al.** [10], the main idea is that the attacker can trick a users web browser into sending a peculiar signal over the Tor network and subsequently detected using traffic analysis. In the paper, they described how a malicious node acting as exit node, when selected by a client can insert a JavaScript code into unencrypted payload and transmit to the client, which then run on client machine and can generate identifiable signal pattern that can be detected by the server. **Evans et al** presented an attack based on legitimate guard node and malicious exit node [11]. In their attack guard node used by a legitimate client is kept busy through the long path generation. This push client to use malicious exit router to relays its traffic **Bauer et al**. [12] demonstrated how extend is the routing selection optimization for performance exposed Tor protocol to end-to-end traffic analysis attack from no global adversaries with minimum resources. Their approach involves compromising guard and exit nodes, by injecting few nodes with high bandwidth and high uptime claims, in a similar work, Bauer et al exploit the role of ports in compromising routing path based on the type of traffic being propagated[13].

Problem Finding

In TOR client (Alice) sends the encrypted message to server (Bob) via several proxies through random way. If attacker want to hack this system or find the message but attacker didn't find path because tor message travel via random path and tor has malicious of exit and entry nodes. So, Attacker knows only exit and entry node and try to inject it. The major drawback of TOR system are describe given below-

- Number of ports is injected via attackers.
- DNS are leaked easily in the time duration of whole communication between client and server and many users can't realise these types of hazard. So, attacker takes advantages of this system when it's reached the original IP of client and done much crime through other identity.

Proposed Solution

We try to do new research in the world of cyber crime for rectified the problem which we are discussed in above section. We are going to introducing the HONEYPOT in a TOR browser. Honey pot is a trap set to detect, deflect, or, in

some manner, counteract attempts at unauthorized use of information systems. Generally, a honey pot consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated and monitored, and which seems to contain information or a resource of value to attackers. This is similar to the police baiting a criminal and then conducting undercover surveillance. Our proposed algorithm done in three phase which describe below:

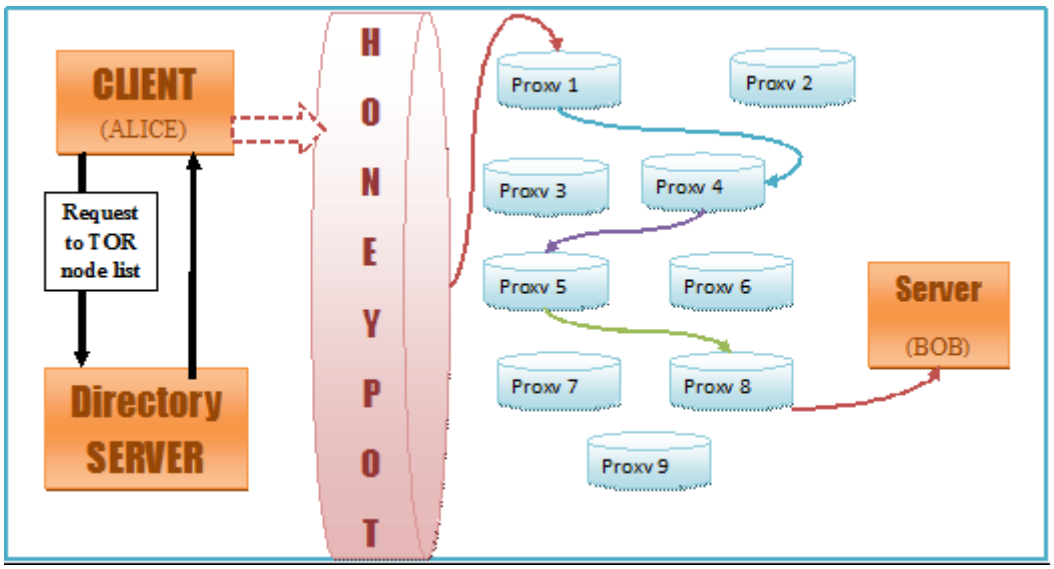

Chart 1: Flow Chart of Proposed Method

PHASE-1 Communication Phase

Step 1

In TOR firstly, client request to directory server for the list of TOR nodes. Then, directory server provides the list of tor nodes to client.

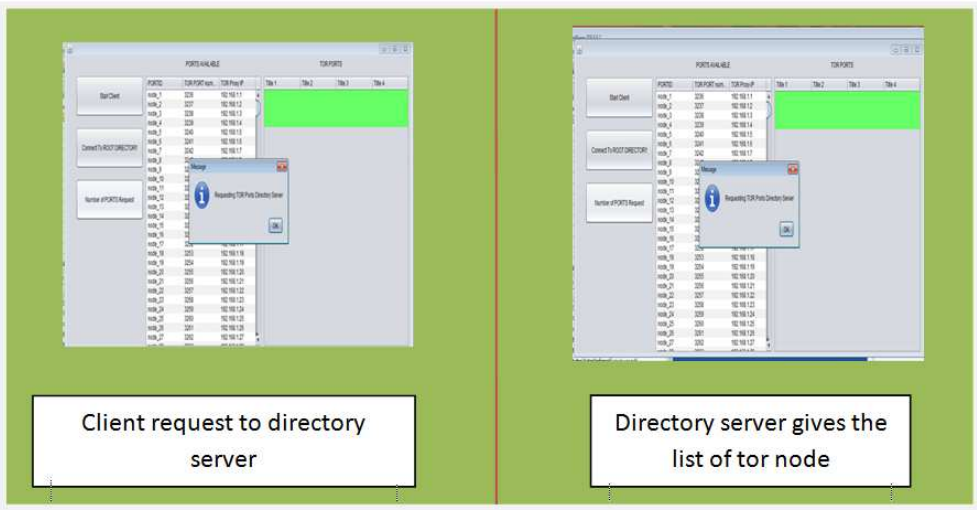

Figure 1

Step 2

Client send encrypted message through tor nodes (proxy server) via random path to the server. In TOR used Diffie-hellman encryption technique for message security.

Phase-2 Attacker Phase

If intruder want to attack the system so firstly he observe the whole system(communication between client and server). He analysed the communication done via random path so its doesn't known the path of communication, its only know the exit and entry node of the system so its try to inject the exit or entry nodes.

Phase 3 Honeypot Technique

Honey pot is a trap set to detect, deflect, or, in some manner, counteract attempts at unauthorized use of information systems. Generally, a honey pot consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated and monitored, and which seems to contain information or a resource of value to attackers. This is similar to the police baiting a criminal and then conducting undercover surveillance.

RESULT ANALYSIS

In this section we compare the existing and proposed technique through some parameters such as port numbers, port injected, IP leak status. We can see our proposed technique are best because we can prevent IP.

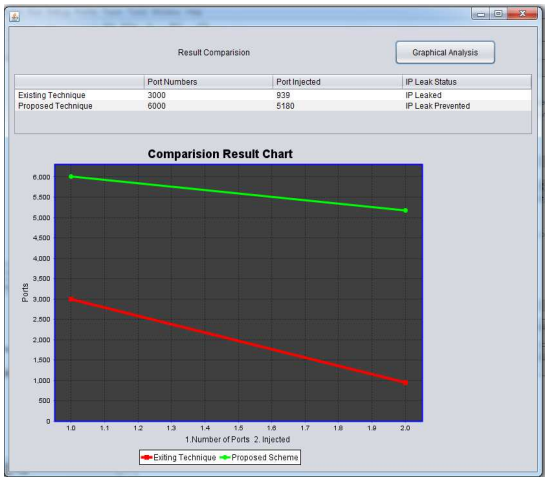

Figure 2

CONCLUSIONS

From our analysis we can say that all tested plugging can be used to disclose the real IP address of the user. Finally we conclude that this technique can reduce the attack in lasting time. This attack is difficult to detect and is able to quickly and accurately confirm the anonymous communication relationship among users on Tor. Due to Tor's fundamental design, defending against this attack remains a very challenging task that we will investigate in our future research. We found that increased traffic spikes within the global DNS for .onion requests corresponded with external global events, emphasizing the potential human factor in those leakages (i.e., user error). While the root cause of these leaked DNS queries remains unknown, our preliminary exploration unveiled concerns to the severity of the leakage and to the possibility of more sensitive private information being unintentionally disclosed. Our future work will continue the examination of leaked DNS queries to the root but will also extend to other non-delegated TLDs such as i2p and .exit. We will plan to further dissect the impact of global events and the role of malware in the leakage, and investigate the potential privacy consequences of the leakage under the various leakage causes. By sharing this introductory work, we wish to trigger further discussion in the community.

REFERENCES

1. Muhammad Aliyu Sulaiman and Sami Zhioua, "Attacking Tor through Unpopular Ports," 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops
2. Jian Bao, Chang-peng Ji and Mo Gao," Research on network security of defense based on Honeypot", 2010 International Conference on Computer Application and System Modeling (ICCASM 2010).
3. John Barker, Peter Hannay, Patryk Szewczyk, "Using traffic analysis to identify The Second Generation Onion Router", 11 Ninth IEEE/IFIP International Conference on Embedded and Ubiquitous Computin.
4. R. Dingledine, N. Mathewson, and P. Syverson, "Tor : the second-generation onion router," in *Proceedings of the 13th Usenix Security Symposium*, August 2004.
5. G. Danezis. Statistical disclosure attacks. In *Security and Privacy in the Age of Uncertainty (SEC2003)*, pages 421–426, Athens, May 2003. IFIP TC11, Kluwer.
6. D. Kesdogan, D. Agrawal, and S. Penz. Limits of anonymity in open environments. In F. Petitcolas, editor, *Information Hiding Workshop (IH 2002)*. Springer-Verlag, LNCS 2578, October 2002.
7. A. Chaabane, P. Manils, and M. Kaafar, "Digging into anonymous traffic: A deep analysis of the tor anonymizing network," in *Proceedings of the 2010 Fourth International Conference on Network and System Security*, ser. NSS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 167–174.
8. Aaron Johnson and Paul Syverson, "More Anonymous Onion Routing Through Trust" 2009 22nd IEEE Computer Security Foundations Symposium.
9. S. J. Murdoch and R. N. Watson, "Metrics for security and performance in low-latency anonymity systems," in Proceedings of the 8th international symposium on Privacy Enhancing Technologies, ser. PETS '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 115–132.
10. T. G. Abbott, K. J. Lai, M. R. Lieberman, and E. C. Price, "Browser-based attacks on tor," in Proceedings of the 7th international conference on Privacy enhancing technologies, ser. PET'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp.184–199.
11. N. Evans, R. Dingledine, and C. Grothoff, "A practical congestion attack on tor using long paths," in Proceedings of the 18th USENIX Security Symposium, August 2009.
12. K. Bauer, D. McCoy, D. Grunwald, T. Kohno, and D. Sicker, "Low-resource routing attacks against Tor," in Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2007), Washington, DC, USA, October 2007.
13. K. Bauer, D. Grunwald, and D. Sicker, "Predicting tor path compromise by exit port," in Proceedings of the 2nd IEEE International Workshop on Information and Data Assurance (WIDA), Phoenix, USA, December 2009.